INTERFACE STRATEGY

Department of Transportation

DELPHI Program

Author:	DELPHI Program Management Team
Creation Date:	March 10, 1998
Last Updated:	5/6/98 4:07 PM
Control Number:	CI040
Version:	5

Contents

Introduction

Purpose

The purpose of this document is to state the strategy for meeting the interface requirements for the DELPHI Program. The Interface Strategy provides a road map for performing the interfacing of data from external systems to the new DELPHI system as well as interfacing data from DELPHI to external systems. The approaches to fulfill this strategy are also presented in this document.

The interface strategy uses the existing Oracle Financials open interfaces which are provided with the Oracle Financials products. Oracle Financials open interfaces are powerful, flexible tools that allow the DELPHI Program team to capture data from external applications, define necessary format conversions, and direct data to the DELPHI system.

The Interface Strategy is used as follows:

- Defining the strategy for successfully interfacing data from external systems to the new DELPHI system.
- Explaining the strategy for successfully interfacing data from DELPHI to external systems.
- Providing guidance for developing and implementing data interfaces.

The Interface Strategy will be provided to every member of the DELPHI Program team.

Scope and Application

The Interface Strategy provides direction for all phases of the data interface effort including design, development and validation. This strategy encompasses those data interface activities necessary for the Global Design and Build, and subsequent OA deployments.

Interface Approach

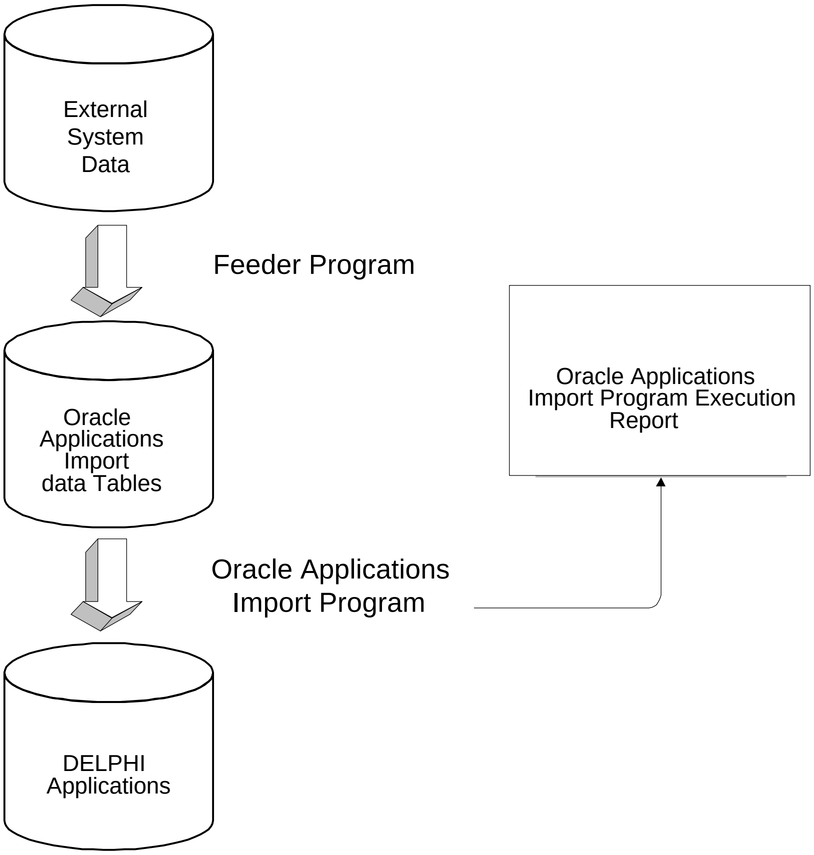
Two approaches will be employed for the interfacing data with DELPHI. They are:

- interfacing external system data to DELPHI via an Oracle open interface; or
- interfacing DELPHI data to an external system or federal entity such as the U.S. Treasury.

The following diagrams and approaches outlines each type of interface.

Oracle Open Interface Approach Diagram

This section provides a graphical representation of the Oracle open interface approach that will be used to interface external data into the DELPHI system. An explanation of this strategy follows.


External System Data
Feeder Program
Oracle Applications Import data Tables
Oracle Applications Import Program Execution Report
Oracle Applications Import Program
DELPHI Applications

Oracle Open Interface Approach

1. Operating Administrations develop feeder programs to meet Oracle Open Interface Format

As identified in the DELPHI Program Charter, all activities involved in the definition, analysis, and development of feeder programs which map external interface data to the appropriate Oracle application import table is the responsibility of the Operating Administrations (OA's). All information required by the OA's to create these feeder programs will be provided by the DELPHI team. This information includes the format of the specific Oracle Application Import Table which will be populated by the OA external interface and definitions of the import table data fields.

The single exception to this policy is the current DOT payroll interface. The activities required to create the feeder program for the payroll data will be the responsibility of the DELPHI team.

2. Populate Oracle Applications Import Table

As described above, the Operating Administrations will create feeder programs to reformat external data to meet the requirements of the appropriate Oracle Applications import table. An example of an Oracle applications import table is the GL_INTERFACE table.

3. Execute the Oracle Application Import Program

Once the data has been reformatted to the proper data structure for the desired Oracle application import table, an Oracle Import Program will be executed to convert the import data into the desired application entries for posting into the DELPHI application.

4. Testing

Unit and Link testing has been integrated into this approach so that the interfaces are tested individually, as well with other business processes within the DELPHI system. This testing ensures that the data interfaced into DELPHI is comprehensive and accurate.

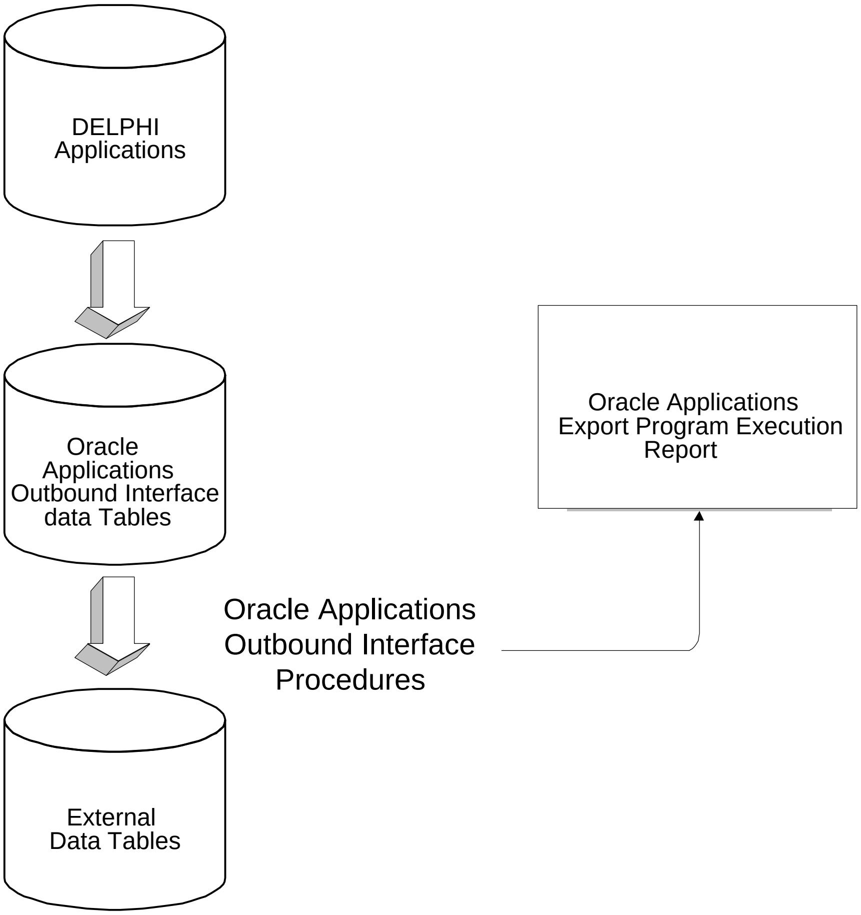
In addition, this approach leverages many standard reports available in the legacy and target system for data validation as well as the Import Program Execution Reports. If no reports support the validation requirements, then custom reports will be created for specific validation purposes.

5. Write and Perform Interface Execution Plan

The interface execution plan is the execution document which will be followed when performing the actual interface. This document is specifically tailored for the DOT data interfaces.

Oracle Outbound Interface Approach Diagram

This section provides a graphical description of the Oracle interface approach that will be used to interface data from DELPHI to other external systems An explanation of this strategy follows.


DELPHI Applications
Oracle Applications Outbound Interface data Tables
Oracle Applications Outbound Interface Procedures
Oracle Applications Export Program Execution Report
External Data Tables

Oracle Outbound Interface Approach

All outbound interfaces generated from DELPHI will be based on existing processes within the Oracle applications. These processes include all standard Federal Government outbound interfaces such as the Treasury Payment Interface and OPAC interface.

1. Execute the Oracle Outbound Interface Process

The process generates an output file with the appropriate file format to meet the requirements of the external system it is feeding. For example, Oracle Federal Payables will generate a payment output file to pay Automated Clearing House Vendors. Once the concurrent process is complete, an output file is generated with the format of “USERID.requested”. This file is stored in the user’s designated output directory in the Concurrent Manager.   A designated system administrator then transmits this data using the appropriate software to the external system (i.e., GOALS).

2. Testing

Unit and Link testing is integrated into this process in order for the interfaces to be tested individually, as well with other business processes within the DELPHI system. This testing ensures that the data interfaced from DELPHI is comprehensive as well as accurate.

In addition, this approach uses many of the standard execution reports available in the DELPHI system such as the Outbound Interface Program Execution Reports. In addition, validation reports are generated for a number of the federal government interface processes, such as the Treasury Payment verification report or OPAC verification. If no reports support the validation requirements, then reports are created for specific validation purposes.

3. Write and Perform Interface Execution Plan

The interface execution plan is the execution document used when performing the actual interface. This document is tailored for the DOT data interfaces.

Automated Tools

OA’s developing feeder programs to reformat their external interface data to meet the Oracle Interface Import Table requirements can use automated tools to assist them. These tools extract data from existing application system’s printed reports, flat file, relational database, or other repository of application information. An example of an automated tool is SQL*Loader. SQL*Loader allows the user to map elements of a regularly formatted file, such as a printed report or flat file, and specify which columns of which tables to populate.

If more complex computations are needed to extract or rearrange information from an external system, a variety of programming languages to populate an import table are available. For example, any of the Oracle–enhanced programming languages, such as Pro*C, Pro*COBOL, or Pro*FORTRAN, as well as SQL*Report can satisfy these needs.

Interface Execution Plan

Each interface process will have an interface execution plan which outlines the data mapping requirements and specific steps necessary to execute the interface. The Interface Execution Plan must include:

- Interface Process Flow Diagram
- Identification of source and target files
- Data element mapping from the external system to DELPHI systems
- Data Translation requirements
- Interface Program Description
- Interface program unit test execution and validation plan

.